

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-121102

(43)Date of publication of application : **12.05.1995**

(51)Int.CI.

G05B 19/05

G06F 9/45

(21)Application number : **05-269197**

(71)Applicant : **OMRON CORP**

(22)Date of filing : 27.10.1993

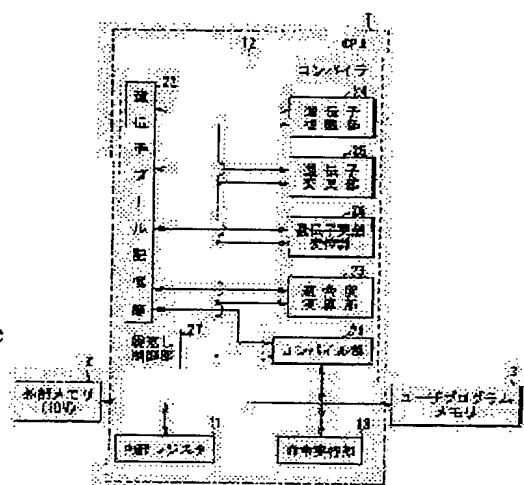
(72)Inventor : **ISHIKAWA HITOSHI**

(54) PROGRAMMABLE CONTROLLER

(57)Abstract:

PURPOSE: To shorten the program execution time by employing genetic algorithm for the scheduling process of a register file and decreasing the frequency of memory access.

CONSTITUTION: A gene is constituted of a register file number sequence based upon the replacement order of a register file. A compiler judges whether or not data required for the execution of each instruction read in when a source program is compiled are stored in an internal register at the time of its execution, and accesses an external memory and outputs an object instruction to replace corresponding data to the internal register on the basis of the gene unless the data are stored. The frequencies of access to the external memory are calculated, gene by gene, and adaptability is found on the basis of the frequencies; and respective genes are duplicated according to the adaptability, and the genes after the duplication are crossed. Some of register file numbers of the genes after the crossing are mutated and this series of processes is repeated for specific generations.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

BEST AVAILABLE COPY

Copyright (C); 1998,2003 Japan Patent Office

(11)特許出願公開番号

特開平7-121102

(43)公開日 平成7年(1995)5月12日.

(51) Int.Cl.⁶

識別記号

室内整理番号

FI

技術表示箇所

G 0 9 B 19/05

G 0 6 F 9/45

7618-3H

9292-5B

G 0 5 B 19/ 05

G 0 6 F 9/ 44

F

3 2 2 H

審査請求 未請求 請求項の数1 OL (全 11 頁)

(21)出願番号 特願平5-269197

(22)出願日 平成5年(1993)10月27日

(71)出願人 000002945

オムロン株式会社

京都府京都市右京区花園土堂町10番地

(72)発明者 石川 仁

京都府京都市右京区花園土堂町10番地 才

ムロン株式会社内

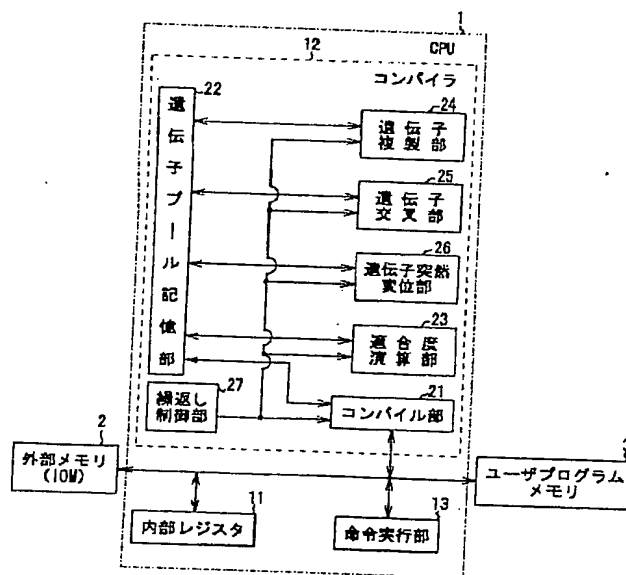
(74) 代理人 弁理士 和田 成則

(54) 【発明の名称】 プログラマブルコントローラ

(57) 【要約】

【目的】 レジスタファイルのスケジューリング処理に遺伝的アルゴリズムを取り入れてメモリアクセス回数を減少させ、プログラム実行時間を短縮化する。

【構成】 遺伝子をレジスタファイルの入替順位に基づいたレジスタファイル番号列で構成する。コンパイラ 12 では、ソースプログラムのコンパイル時に読込まれた各命令についてその命令実行に必要なデータが内部レジスタ 11 に格納されているか否かを判断して、格納されていない場合には外部メモリ 2 へアクセスして当該データを遺伝子に基づいて内部レジスタ 11 に入替えるオブジェクト命令を出力する。そして、遺伝子毎に外部メモリ 2 へのアクセス回数を算出してその回数に基づき適合度を求め、各遺伝子をその適合度に応じて複製し、さらに複製後の遺伝子を交叉すると共に、その交叉後の遺伝子のレジスタファイル番号の幾つか突然変異させ、このような一連の処理を所定世代繰り返す。



【特許請求の範囲】

【請求項 1】 ソースプログラム中の各命令をコンパイルしてそのオブジェクト命令を出力すると共に、内部レジスタに格納されるデータをスケジューリング手順に基づいて入替えるオブジェクト命令を出力するコンパイラを備えたプログラマブルコントローラであって、上記コンパイラは、

スケジューリング手順を示す内部レジスタのレジスタファイル番号列を遺伝子として複数記憶する遺伝子プール記憶手段と、

ソースプログラム中の各命令を順次読み込んで、その命令のオブジェクト命令を出力すると共に、その命令実行の際に必要とされるデータが内部レジスタのレジスタファイルに格納されているか否かを判断して、格納されていない場合には上記遺伝子プール記憶手段に記憶された各遺伝子に基づいてレジスタファイルを指定し、当該レジスタファイルへ上記データが格納されるよう外部メモリへアクセスするオブジェクト命令を出力するコンパイル手段と、

上記コンパイル手段から出力される外部メモリにアクセスするオブジェクト命令の数を遺伝子毎に算出して、その命令数に基づき各遺伝子の適合度を求める適合度演算手段と、

上記適合度演算手段によって求められた適合度に応じて各遺伝子を複製する遺伝子複製手段と、

上記遺伝子複製手段によって複製された各遺伝子を交叉させ新たな遺伝子を作成する遺伝子交叉手段と、

上記遺伝子交叉手段によって作成された新たな遺伝子を構成するレジスタファイル番号の幾つかを任意の確率で任意のレジスタ番号に突然変異させる遺伝子突然変異手段と、

上記コンパイル手段、上記適合度演算手段、上記遺伝子複製手段、上記遺伝子交叉手段および上記遺伝子突然変異手段に各々の処理を繰返し所定回数実行させ、所定回数実行後に各々の処理を停止させる繰返し制御手段と、を具備することを特徴とするプログラマブルコントローラ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、ソースプログラム中の各命令をコンパイルしてそのオブジェクト命令を出力すると共に、内部レジスタに格納されるデータをスケジューリング手順に基づいて入替えるオブジェクト命令を出力するコンパイラを備えたプログラマブルコントローラに関する。

【0002】

【従来の技術】 ロードストア方式の CPU を利用したプログラマブルコントローラ（以下、PLC という）においては、通常、外部メモリへのアクセス命令実行時間が内部レジスタへのアクセス命令実行時間よりも非常に長

いため、外部メモリへのアクセス回数が多いプログラムの場合、この部分の実行時間がプログラム全体へ与える影響が非常に大きくなる。

【0003】 従って、外部メモリへのアクセス回数を減らしてプログラム実行時間の短縮化を図るために CPU 内部のレジスタ（以下、レジスタという）のレジスタファイルを有効利用することが不可欠で、レジスタファイルへ 1 度あるデータを読み込んでおけば、その後そのデータへのアクセスはすべてレジスタファイルに対して行うことが提案されている。

【0004】

【発明が解決しようとする課題】 しかし、レジスタファイルは有限でそれほど容量がないため、ユーザプログラムに合わせてレジスタファイルのスケジューリング、すなわちファイルの内容を入替える必要があるが、どのようなスケジューリング手順でレジスタファイルを入替れば外部メモリへのアクセス回数を最も減少できるのかわ明かにされていない。

【0005】 また、最近、情報処理の分野では、問題解決の一つの手法として遺伝的アルゴリズム（GA ; Genetic Algorithm）の導入が提案されている。

【0006】 そこで、本発明は、このような問題に着目してなされたもので、レジスタファイルのスケジューリングに遺伝的アルゴリズムを取り入れることによって外部メモリへのアクセス回数を減少させて、プログラム実行時間を短縮化することができる PLC を提供することを目的とする。

【0007】

【課題を解決するための手段】 上記目的を達成するため、本発明では、ソースプログラム中の各命令をコンパイルしてそのオブジェクト命令を出力すると共に、内部レジスタに格納されるデータをスケジューリング手順に基づいて入替えるオブジェクト命令を出力するコンパイラを備えたプログラマブルコントローラであって、上記コンパイラは、スケジューリング手順を示す内部レジスタのレジスタファイル番号列を遺伝子として複数記憶する遺伝子プール記憶手段と、ソースプログラム中の各命令を順次読み込んで、その命令のオブジェクト命令を出力すると共に、その命令実行の際に必要とされるデータが内部レジスタのレジスタファイルに格納されているか否かを判断して、格納されていない場合には上記遺伝子プール記憶手段に記憶された各遺伝子に基づいてレジスタファイルを指定し、当該レジスタファイルへ上記データが格納されるよう外部メモリへアクセスするオブジェクト命令を出力するコンパイル手段と、上記コンパイル手段から出力される外部メモリにアクセスするオブジェクト命令の数を遺伝子毎に算出して、その命令数に基づき各遺伝子の適合度を求める適合度演算手段と、上記適合度演算手段によって求められた適合度に応じて各遺伝子を複製する遺伝子複製手段と、上記遺伝子複製手段に

よって複製された各遺伝子を交叉させ新たな遺伝子を作成する遺伝子交叉手段と、上記遺伝子交叉手段によって作成された新たな遺伝子を構成するレジスタファイル番号の幾つかを任意の確率で任意のレジスタ番号に突然変異させる遺伝子突然変異手段と、上記コンパイル手段、上記適合度演算手段、上記遺伝子複製手段、上記遺伝子交叉手段および上記遺伝子突然変異手段に各々の処理を繰返し所定回数実行させ、所定回数実行後に各々の処理を停止させる繰返し制御手段と、を具備することを特徴とする。

【0008】

【作用】本発明に係るコンパイラでは、スケジューリング手順を示す内部レジスタのレジスタファイル番号列を遺伝子として複数記憶し、ソースプログラム中の各命令を順次読み込んで、その命令のオブジェクト命令を出力すると共に、その命令実行の際に必要なとされるデータが内部レジスタのレジスタファイルに格納されているか否かを判断して、格納されていない場合には上記遺伝子プール記憶手段に記憶された各遺伝子に基づいてレジスタファイルを指定し、当該レジスタファイルへ上記データが格納されるよう外部メモリへアクセスするオブジェクト命令を出力する。

【0009】そして、外部メモリにアクセスするオブジェクト命令の命令数を遺伝子毎に算出して、その命令数に基づき各遺伝子の適合度を求めて、適合度に応じて各遺伝子を複製し、続いて複製された各遺伝子を交叉させ新たな遺伝子を作成すると共に、新たな遺伝子を構成するレジスタファイル番号の幾つかを任意の確率で任意のレジスタ番号に突然変異させる。このような処理を繰返し所定回数実行させ、所定回数実行後に各処理を停止する。

【0010】

【実施例】以下、本発明に係るPLCの実施例について説明する。

【0011】図1は、本発明に係るPLCの実施例の構成を示している。

【0012】このPLCは、CPU1と、入出力データを格納した外部メモリ(I/O)2と、ラダー表記等でユーザが作成したソースプログラムおよびそのオブジェクトプログラムが格納されるユーザプログラムプログラムメモリ3とを有する。

【0013】CPU1は、内部レジスタ11と、ユーザが作成したソースプログラムのコンパイルを行ってそのオブジェクト命令を出力すると共に、遺伝的アルゴリズム(GA)によって後述するレジスタファイルのスケジューリングを行うコンパイラ12と、オブジェクト命令を外部メモリ2または内部レジスタ11にアクセスしながら実行する命令実行部13とを有している。

【0014】内部レジスタ11は、例えばワード単位等のデータ格納エリアであるレジスタファイルを複数有し

ており、外部メモリ2に記憶されたデータの内の一部をそのレジスタファイルに格納するように構成されている。

【0015】コンパイラ12は、その機能がブロックで示されており、コンパイル部21と、遺伝子プール記憶部22と、適合度演算部23と、遺伝子複製部24と、遺伝子交叉部25と、遺伝子突然変異部26と、繰返し制御部27とから構成されている。

【0016】コンパイル部21は、ソースプログラム中の各命令を順次読み込んで、その命令のオブジェクト命令を出力すると共に、詳細は後述するがその命令実行の際に必要なとされるデータが内部レジスタのレジスタファイルに格納されているか否かを判断して、格納されていない場合には後述する遺伝子プール記憶部22に記憶された各遺伝子に基づいてレジスタファイルを指定し、当該レジスタファイルへ上記データが格納されるよう外部メモリ2へアクセスするオブジェクト命令を出力するように構成されている。

【0017】また、遺伝子プール記憶部22は、スケジューリング手順を示す内部レジスタ11のレジスタファイル番号列を遺伝子として複数記憶しており、適合度演算部23は、コンパイル部21から出力される外部メモリ2にアクセスするオブジェクト命令の数を遺伝子毎に算出して、その命令数に基づき各遺伝子の適合度を求めるように、遺伝子複製部24は、適合度演算部24によって求められた適合度に応じて各遺伝子を複製するように構成されている。

【0018】さらに、遺伝子交叉部25は、遺伝子複製部24によって複製された各遺伝子を交叉して新たな遺伝子を作成するように、遺伝子突然変異部26は、遺伝子交叉部25によって作成された新たな遺伝子を構成するレジスタファイル番号の幾つかを任意の確率で任意のレジスタ番号に突然変異させるように、繰返し制御部27は、上記処理部21～26に各々の処理を繰返し所定回数実行させ、所定回数実行後に各々の処理を停止させるように構成されている。

【0019】なお、本実施例では、コンパイラ12は便宜上CPU1内部に設けて説明するが、本発明では、CPU1内部でなく、CPU1外部のツール(図示せず)等にそのコンパイラ12の機能を持たせ、ツールがユーザの作成したソースプログラムをコンパイルすると共にそのオブジェクトプログラムを直接ユーザプログラムメモリ3に転送し、ユーザプログラムメモリ3がオブジェクトプログラムのみを格納するような構成にしても良い。

【0020】図2は、本実施例で実行される遺伝的アルゴリズムによるレジスタファイルのスケジューリング処理で採用される遺伝子の構造を示している。

【0021】遺伝子は、図に示すように、“2, 5, 3, 6, 1, . . .”という内部レジスタ11のデータ

格納単位を示すレジスタファイルの入替順位に基づいて配列されたレジスタファイル番号列から構成されており、この場合には、“2”の番号のレジスタファイルから最初に入替えられることを示している。

【0022】次に、コンパイラ12における遺伝的アルゴリズムによるレジスタファイルのスケジューリング処理を説明する。

【0023】まずは、このスケジューリング処理中で適合度を計算するために必要なデータについて説明する。

【0024】図3に、適合度計算のために必要なデータを示す。

【0025】このデータには、スケジュール管理用テーブルTに格納されるレジスタファイルの制御データと、プログラム実行時の外部メモリ2へのアクセストータル回数を示すメモリアクセストータル回数 (MEMACC) とがある。

【0026】レジスタファイルの制御データには、内部レジスタ11の格納領域である各レジスタファイルのレジスタファイル番号R0, R1, …と対応して3つの制御データ、すなわち当該レジスタファイルに格納されるデータのチャンネルアドレス (table[].address) と、当該レジスタファイルが使用されているか否かを示す使用／非使用フラグ (table[].on_use) と、当該レジスタファイルの内容が外部メモリ2の内容と変化しているか否かを示す変更済みフラグ (table[].modified) とがある。なお、使用／非使用フラグ (table[].on_use) は、使用されているとき“1”がセットされ、非使用のときは“0”がセットされる。また、変更済みフラグ (table[].modified) は、レジスタファイルの内容が外部メモリ2の内容と変更されていなければ“0”がセットされ、外部メモリ2の内容から変更された場合には“1”がセットされることになる。

【0027】次に、遺伝的アルゴリズムによるレジスタファイルのスケジューリング処理を説明する。

【0028】図4は、このレジスタファイルのスケジューリング処理の全体を示している。この処理では、遺伝子を構成する各レジスタファイル番号の順位を“j”で示し、遺伝子[j]という場合には、j番目のレジスタファイル番号を示すものとする。また、ラダーソースプログラムを構成する各命令の順位を“i”で示し、命令[i]という場合には、i番目の命令を示すものとする。

【0029】まず、適合度を計算する場合には、図3で説明したレジスタファイルにおける制御用データの初期化処理を後述する図5に示すように行い (ステップ200)、続いてラダーソースプログラム中の命令の順位を示すiに“0”をセットすると共に (ステップ210)、遺伝子中のレジスタファイル番号の順位を示すjに“0”をセットする (ステップ220)。

【0030】そして、コンパイル部21がユーザプログ

ラムメモリ3から順次ラダーソースプログラムの命令[i]の読み込みを行い (ステップ230)、その際データ入替え部23が当該命令[i]についてレジスタファイルの入替えの必要があるか否か、すなわち当該命令[i]実行の際に必要なチャンネルデータが内部レジスタ11のレジスタファイルに格納されているか否かを後述する図6に示すように判断する (ステップ240)。

【0031】ここで、レジスタファイルの入替え必要有りと判断された場合には (ステップ240 “Yes”)、遺伝子[j]のレジスタファイル番号のレジスタファイルを入替え対象として入替えを後述する図7に示すように行くと共に (ステップ250)、jの値を1インクリメントして (ステップ260)、続いて次の命令を指定するためiの値を1インクリメントする (ステップ270)。その一方、レジスタファイルの入替え必要なしと判断された場合には (ステップ240 “No”)、レジスタファイルの入替えも、jの値のインクリメントも行わずに、iの値を1インクリメントする (ステップ270)。

【0032】そして、レジスタファイルに記憶されたデータを読み出して当該命令の評価、すなわち後述する図8に示すように当該命令がO U T系の命令か否かの判断処理およびオブジェクト出力を行い (ステップ280)、続いて当該命令がラダーソースプログラムの終りに達したか否かを判断して (ステップ290)、ラダーソースプログラムの終りに達してない場合には (ステップ290 “No”)、次の新たな命令[i]の読み込みを行って (ステップ230)、上記と同様のステップ240～290の処理を行う。

【0033】一方、ラダーソースプログラムの終りに達した場合には (ステップ290 “Yes”)、この遺伝子が他の遺伝子と比べどれだけ優れているかを示す適合度の計算を後述する図9に示すようにメモリアクセストータル回数 (MEMACC) を基にして行い (ステップ300)、次いでこのようなステップ200～300までの処理を全ての遺伝子について行ったか否かを判断する (ステップ310)。

【0034】ここで、全ての遺伝子について行っていない場合には (ステップ310 “No”)、このようなステップ200～300までの処理を他の遺伝子について繰返すようにする。

【0035】これに対し、全ての遺伝子についてステップ200～300までの処理を行っている場合には (ステップ310 “Yes”)、後述する図10に示すように遺伝子複製部24が各遺伝子をその適合度に応じて複製し (ステップ320)、続いて後述する図11に示すように遺伝子交叉部25が複製後の遺伝子間で交叉を行い (ステップ330)、さらに後述する図12に示すように遺伝子突然変異26が交叉後の新たな遺伝子を構成するレジスタファイル番号の幾つかを任意の確率で任意の

レジスタ番号に突然変異させる(ステップ340)。これで遺伝的アルゴリズムの1世代が終了する。

【0036】そして、遺伝的アルゴリズムの場合には、通常このようなステップ200～340までの処理を所定世代繰り返して解となる遺伝子を進化すなわち改良する必要があるため、繰返し制御部28が1世代終了する度に予め定めた所定の世代数繰り返したか否かを判断して(ステップ350)、その所定世代数まで達してない場合には(ステップ350“N o”)、ステップ200～340までの各遺伝子についての適合度計算から突然変異までの処理を繰り返すようにする一方、所定世代繰り返したと判断した場合には(ステップ350“Ye s”)、レジスタファイルのスケジューリング処理を終了する。

【0037】なお、実際には適合度の計算とコンパイルとは同時に行うことも可能なので、上記コンパイル部21中に適合度演算部23を入れて考えても良く、上記ステップ200～310までの処理をコンパイル部21および適合度演算部23が行うものとする。

【0038】図5は、図4のステップ200に示すレジスタファイルにおける制御用データの初期化処理を詳細に示している。

【0039】この処理では、まずkに初期値“0”をセットして(ステップ300)、チャンネルアドレス(table[k].address)を初期化し(ステップ310)、次いで使用／非使用フラグ(table[k].on_use)を初期化し(ステップ320)、さらに変更済みフラグ(table[k].modified)を初期化して(ステップ330)、kを1インクリメントする(ステップ340)。そして、このような処理をレジスタファイル数分繰り返したか否かを判断して(ステップ350)、レジスタファイル数分繰り返してない場合には(ステップ350“N o”)、次のkに対応した各制御用データについて初期化を行い(ステップ310～330)、レジスタファイル数分繰り返した場合には(ステップ350“Ye s”)、この初期処理を終了する。

【0040】図6は、図4のステップ240に示すレジスタファイル入替え判断処理を詳細に示している。

【0041】この処理では、まずkに初期値“0”をセットして(ステップ400)、使用／非使用フラグ(table[k].on_use)を参照することによりレジスタファイル番号kが使用されているか否かを判断し(ステップ410)、続いてそのチャンネルアドレス(table[k].address)を参照することにより当該アドレスがラダーソースプログラムから読み込んだ命令[i]の実行に必要とされるデータのチャンネルアドレスと一致するか否かの判断を行う(ステップ420)。

【0042】そして、レジスタファイル[k]が使用され(ステップ410“Ye s”)、かつ、そのチャンネルアドレス(table[k].address)が当該命令[i]実行に必

要なデータのチャンネルアドレス(Address)と一致すると判断された場合には(ステップ420“Ye s”)、レジスタファイルの入替え必要なしを示す“0”を返す。

【0043】一方、レジスタファイル[k]が使用されてない場合(ステップ410“N o”)、およびそのレジスタファイル[k]が使用されていてもそのチャンネルアドレス(table[k].address)が当該命令[i]実行に必要なデータのチャンネルアドレス(Address)と一致しないと判断された場合には(ステップ410“Ye s”, ステップ420“N o”)、kの値を1インクリメントして(ステップ440)、以上の処理をレジスタファイル数分繰り返したか否かを判断して(ステップ450)、レジスタファイル数分繰り返してない場合には(ステップ450“N o”)、その新たなkで上記ステップ410、420の判断処理を行うようにする一方、kの値がレジスタファイル数分繰り返した場合に(ステップ450“Ye s”)、当該命令[i]実行に必要なデータがレジスタファイルに格納されてないことを示しているため、レジスタファイルの入替え必要有りを示す“1”を返す。

【0044】図7は、図4のステップ250に示すレジスタファイルの入替え処理を詳細に示している。

【0045】この処理では、まずkを遺伝子[j]のレジスタファイル番号、すなわち遺伝子の先頭からj番目に記述されているレジスタファイル番号にセットし(ステップ500)、続いてそのレジスタファイル番号の使用／非使用フラグ(table[k].on_use)を参照してレジスタファイル[k]が使用されているか否かの判断を行うと共に(ステップ510)、その変更済みフラグ(table[k].modified)を参照してそのレジスタファイル[k]のレジスタファイルの内容が外部メモリ2の内容と変更しているか否かの判断を行う(ステップ520)。

【0046】ここで、レジスタファイル[k]が使用されていて(ステップ510“Ye s”)、かつ、その内容が外部メモリ2の内容から変更されていると判断された場合には(ステップ520“Ye s”)、必要であれば当該レジスタファイル[k]の内容を外部メモリ(I O M)2に書き込むストア系のオブジェクト命令を出力し(ステップ530)、メモリアクセストータル回数(ME MACC)の値を1インクリメントする(ステップ540)。

【0047】一方、レジスタファイル[k]が使用されていない場合(ステップ510“N o”)、およびレジスタファイル[k]が使用されているが、その内容が外部メモリ2の内容から変更していないと判断された場合には(ステップ510“Ye s”, ステップ520“N o”)、当該レジスタファイル内のデータを外部メモリ2へ書き込まないようにする。

【0048】次いで、両場合とも、必要であれば当外部

メモリ (IOM) 2 に格納された当該データをレジスタファイル [k] に読み込むロード系のオブジェクト命令を出力し (ステップ 550)、続いてメモリアクセストータル回数 (MEMACC) の値を 1 インクリメントして (ステップ 560)、当該レジスタファイル [k] の使用/非使用フラグ (table[k].on_use) に使用を示す

“1” をセットすると共に (ステップ 570)、さらにその変更済みフラグ (table[k].modified) に外部メモリ 2 の内容からの変更なしを示す “0” をセットして (ステップ 580)、そのチャンネルアドレス (table[k].address) に当該命令のアドレス (Address) を設定する (ステップ 590)。

【0049】図 8 は、図 4 のステップ 280 に示す命令の評価処理を示している。

【0050】この処理では、読み込んだ当該命令 [i] が OUT 系の命令であるか否かを判断して (ステップ 600)、OUT 系の命令の場合のみ (ステップ 600 “Yes”)、レジスタファイル [k] の変更済みフラグ (table[k].modified) に当該レジスタファイルの内容が外部メモリ 2 の内容から変更されていることを示す “1” をセットし (ステップ 610)、続いてコンパイラ部 21 が命令に応じたオブジェクトの出力を行い (ステップ 620)、この評価処理を終了する。

【0051】図 9 は、図 4 のステップ 300 に示す適合度計算処理の際に使用される関数の一例を示している。

【0052】適合度は、“適合度 = f (MEMACC)”、すなわちメモリアクセストータル回数 (MEMACC) の関数 f で算出される。関数 f としては、メモリアクセストータル回数 (MEMACC) が小さいほど大きい値を返す、すなわちメモリアクセストータル回数 (MEMACC) が小さいほど適合度が大きくなるような関数を使用し、例えば (a) に示すような一次関数や、(b) に示すようなシグモイド関数が挙げられる。図 10 は、図 4 のステップ 320 に示す遺伝子の複製処理を簡単に示している。複製処理では、遺伝子 a、b を適合度に応じ、すなわち適合度の大きいものほど多く作成するようにする。

【0053】図 11 は、図 4 のステップ 330 に示す遺伝子の交叉処理を簡単に示している。交叉処理では、2 つの遺伝子 a、b 間で、遺伝子 a、b を構成するレジスタファイル番号列 “987654”、“123456” を同じ位置 (図では、“^” で示している) で各々分断し、それぞれの前半部 (あるいは後半部) を入替えて、“987456” というレジスタファイル番号列からなる遺伝子 c と、“123654” というレジスタファイル番号列からなる遺伝子 d とを作成するようにする。

【0054】図 12 は、図 4 のステップ 340 に示す遺伝子の突然変異処理を簡単に示している。突然変異処理では、前述の通り遺伝子 a について突然変異を起こさせる場合には、遺伝子 a のレジスタファイル番号列 “123456” のうち例えば任意のビットである 4 ビット目

の “5” を他の任意のレジスタファイル番号である “9” に変更するようにする。

【0055】以上詳細に説明したように、本実施例では、CPU 1 内のコンパイラ 12 は、コンパイラ部 21 によるソースプログラムのコンパイル時に、遺伝的アルゴリズムを使用したレジスタファイルのスケジューリング処理を行って、スケジューリング手順である各遺伝子に基づいてレジスタファイルの入替えを行うと共にその適合度を世代毎に算出し、さらに遺伝子をその適合度に応じて複製、交叉、突然変異と所定世代繰り返して進化させる。

【0056】このため、世代を繰返すにしたがって、適合度の高い、すなわち外部メモリ 2 へのアクセス回数が少ない遺伝子ほど残り、その割合が増え、その遺伝子に基づいてレジスタファイルのスケジューリング処理を行うことになるので、所定世代を繰返した後、命令実行部 11 がコンパイル後のオブジェクトプログラムを実行する際には、外部メモリ 2 へのアクセス回数が減少し、プログラム実行時間を短縮化することが可能になる。

【0057】なお、遺伝子の初期値としてまったくランダムな値から出発した場合に、ある程度優れた遺伝子の値を全遺伝子数の半分だけ初期設定した方法より優れた結果が得られるケースのあることが実験的にわかった。これより、初期状態としてあまりに解としての遺伝子を決定してしまうと、遺伝子の多様性に初めから制限が加えられ、最適解に収束しにくくなる、すなわちそれ以上優れた遺伝子が発生しにくくなることを示しており、レジスタファイルのスケジューリング手順の算出においても、遺伝子には多様なものが必要であることを示している。

【0058】

【発明の効果】以上説明したように、本発明では、レジスタファイルのスケジューリング処理に遺伝的アルゴリズムを取り入れて、スケジューリング手順であるレジスタファイルの入替順位を示すレジスタファイル番号列を遺伝子とし、ソースプログラムのコンパイル時に各遺伝子に基づきレジスタファイルの入替えを行うと共にその適合度を世代毎に算出して、各遺伝子をその適合度に応じて複製、交叉、突然変異させる処理を何世代も繰り返して遺伝子を進化させる。

【0059】このため、世代を繰返すにしたがって、適合度の高い、すなわち外部メモリへのアクセス回数が少ない遺伝子の割合が増え、その遺伝子に基づいてレジスタファイルのスケジューリング処理が行われるので、所定世代後、プロセッサがオブジェクトプログラムを実行する際には、外部メモリへのアクセス回数が減少して、プログラム実行時間を短縮化できる。

【図面の簡単な説明】

【図 1】本発明に係る PLC の実施例の構成を示すブロック図。

11

【図 2】本実施例のスケジューリング処理で採用される遺伝子の構造を示す説明図。

【図 3】適合度計算のために必要なデータを示す説明図。

【図 4】レジスタファイルのスケジューリング処理を示すフローチャート。

【図 5】制御用データの初期化処理を示すフローチャート。

【図 6】レジスタファイル入替え判断処理を示すフローチャート。

【図 7】レジスタファイルの入替え処理を示すフローチャート。

【図 8】命令の評価処理を示すフローチャート。

【図 9】適合度計算処理の際に使用される関数の一例を示す説明図。

【図 10】遺伝子の複製処理を示す説明図。

12

【図 11】遺伝子の交叉処理を示す説明図。

【図 12】遺伝子の突然変異処理を示す説明図。

【符号の説明】

1 CPU (プロセッサ)

2 外部メモリ

3 ユーザプログラムメモリ

11 内部レジスタ

12 コンパイラ

13 命令実行部

10 21 コンパイル部

22 遺伝子プール記憶部 (遺伝子プール記憶手段)

23 適合度演算部 (適合度演算手段)

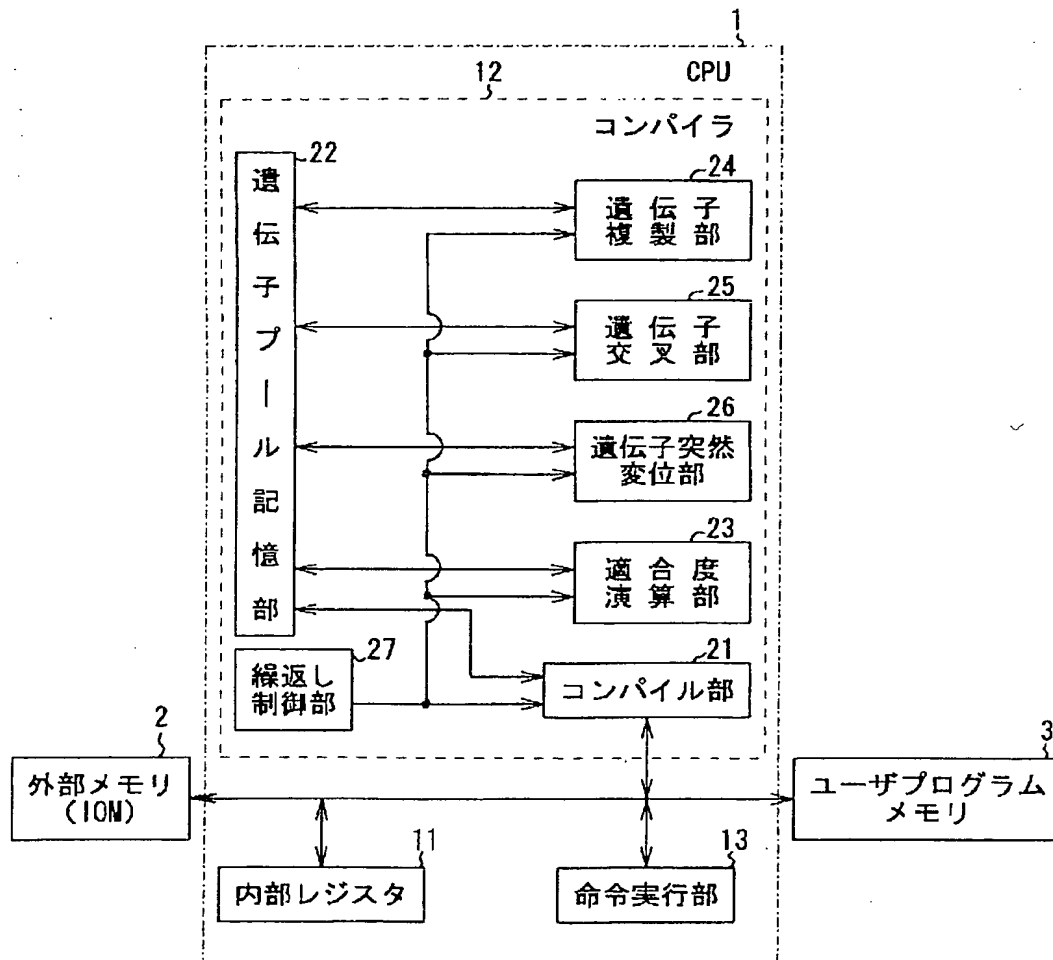
24 遺伝子複製部 (遺伝子複製手段)

25 遺伝子交叉部 (遺伝子交叉手段)

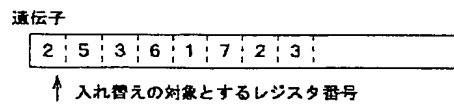
26 遺伝子突然変異部 (遺伝子突然変異手段)

27 繰返し制御部 (繰返し制御手段)

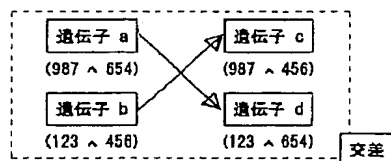
【図 1】



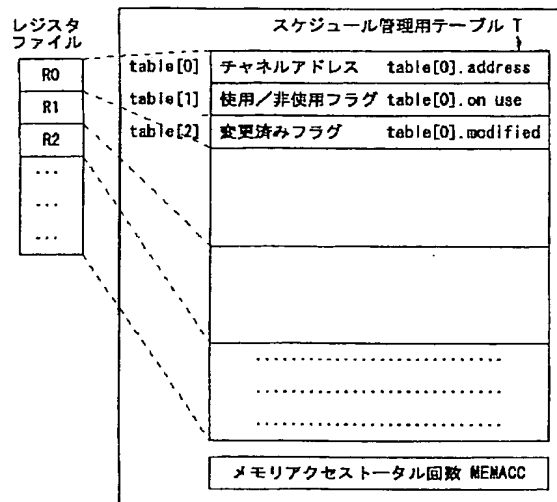
【図 2】



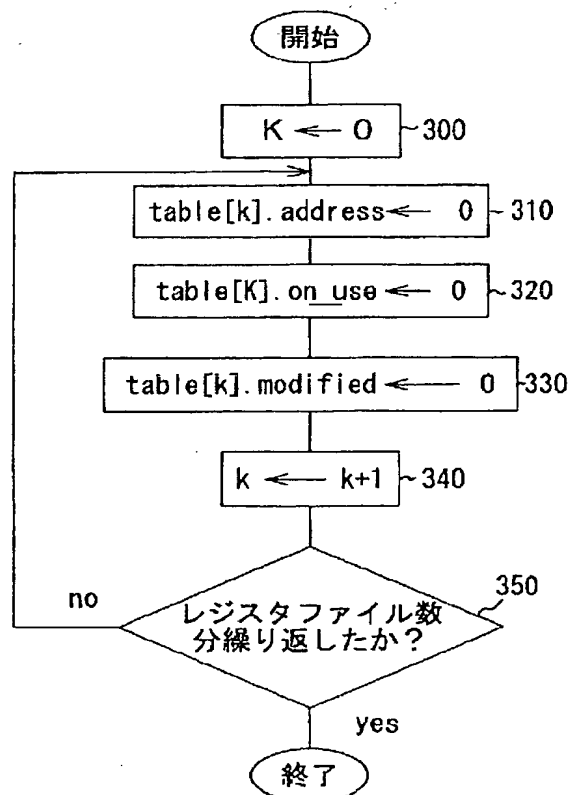
【図 1 1】



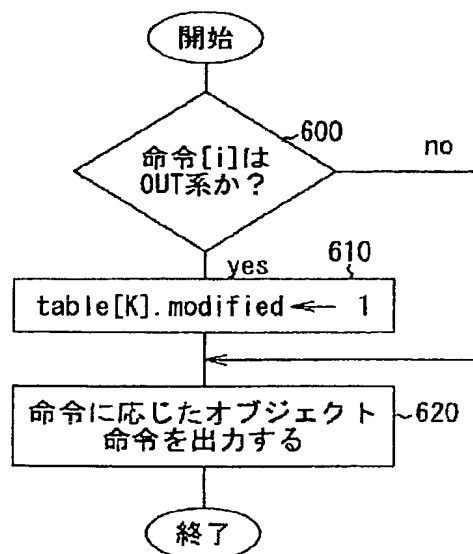
【図 3】



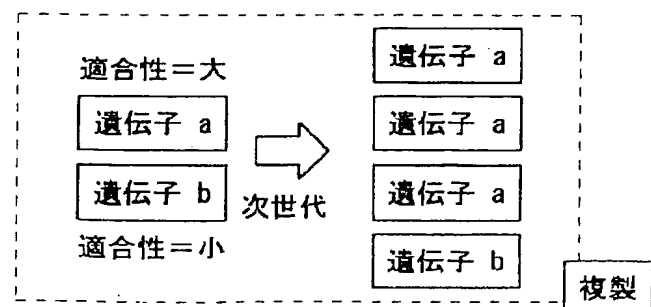
【图 5】



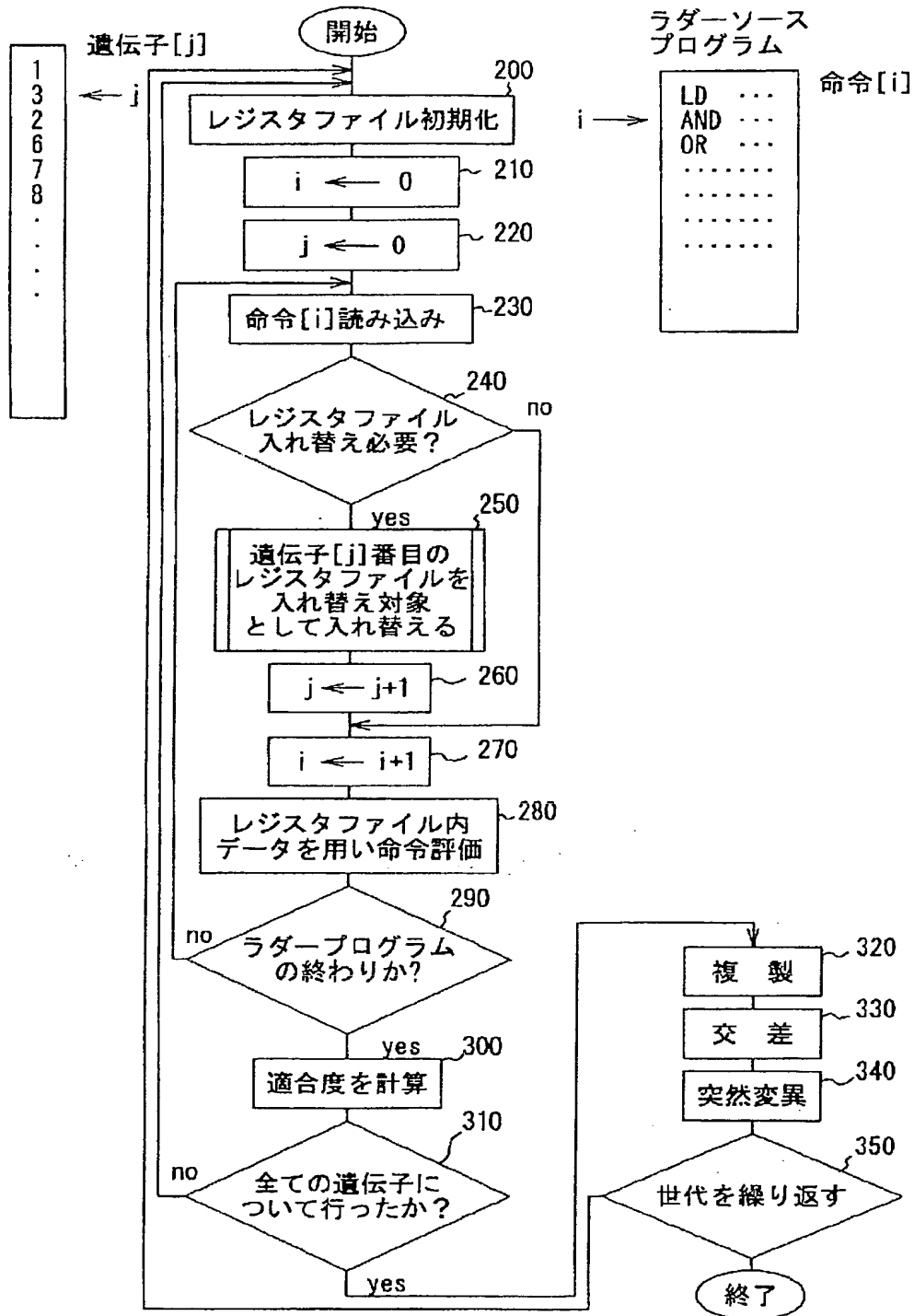
【图 8】



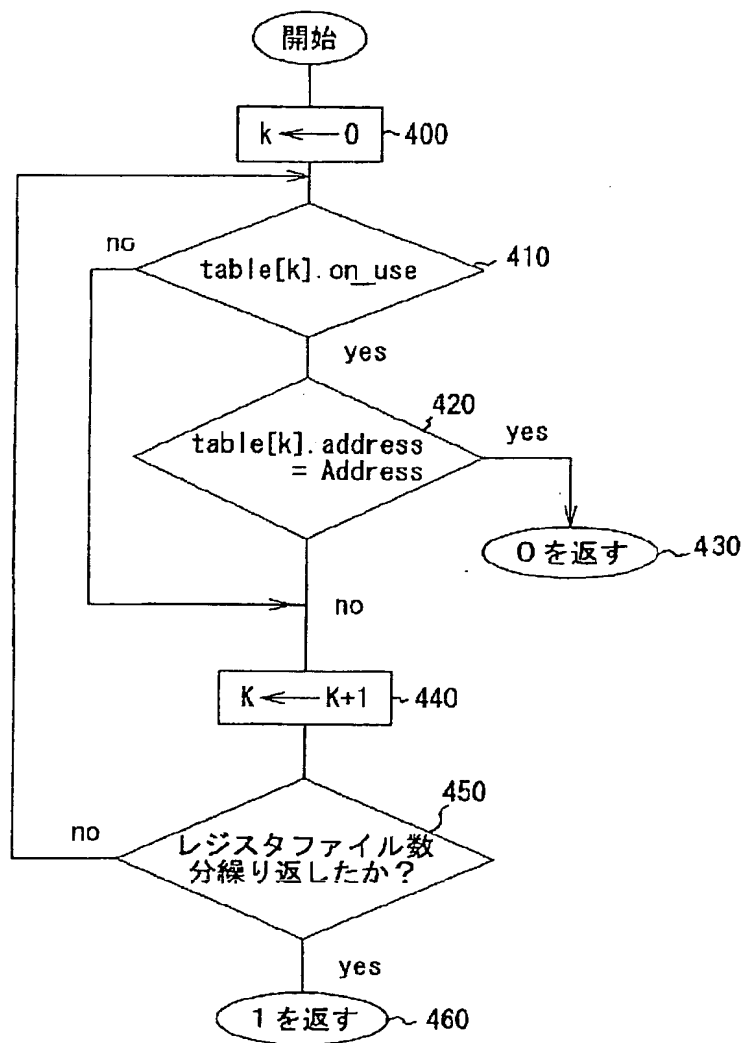
【図 10】



【図 4】



【図 6】

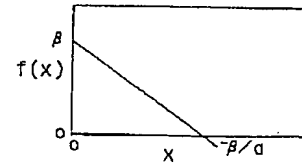


【図 9】

(a)

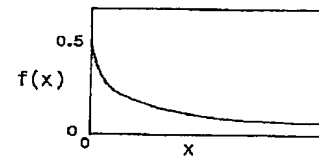
$$f(x) = \alpha x + \beta$$

α, β : 定数

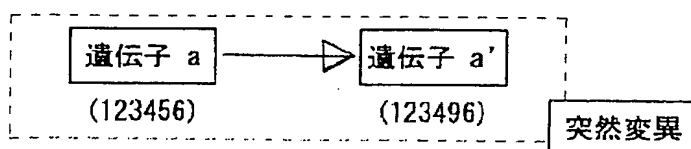


(b)

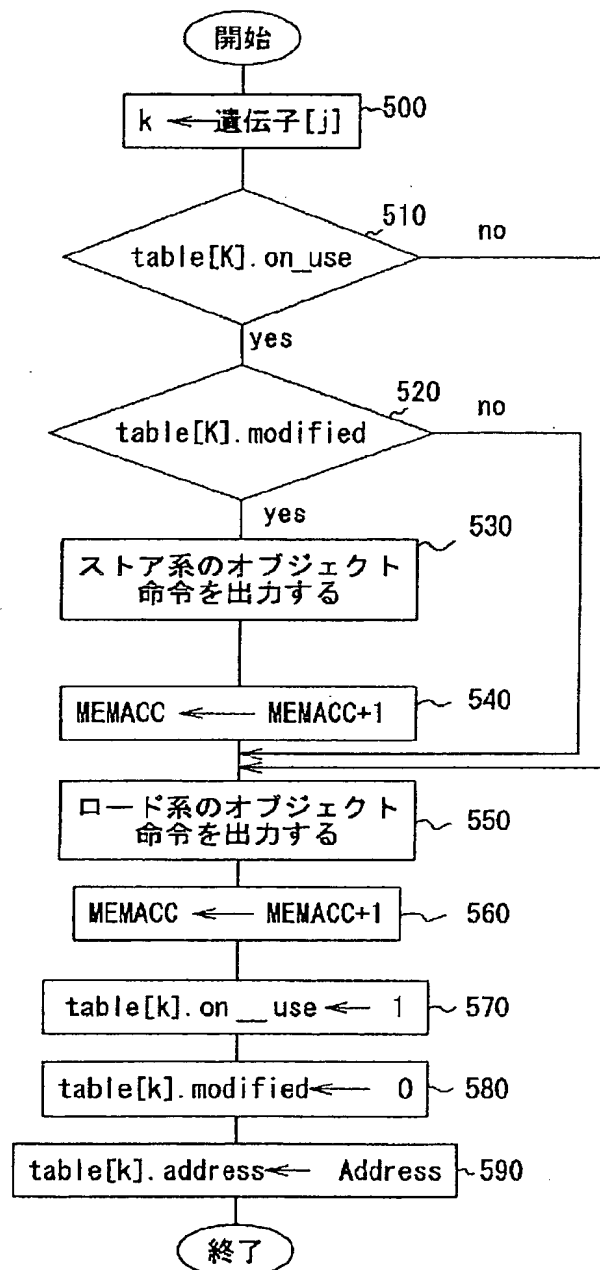
$$f(x) = 1/(1 + \exp(x)) \dots$$



【図 12】



【図 7】



【公報種別】公開特許公報の訂正

【部門区分】第 6 部門第 3 区分

【発行日】平成 8 年（1 9 9 6）4 月 2 日

【公開番号】特開平 7—1 2 1 1 0 2

【公開日】平成 7 年（1 9 9 5）5 月 1 2 日

【年通号数】公開特許公報 7—1 2 1 2

【出願番号】特願平 5—2 6 9 1 9 7

【訂正要旨】分類誤載につき下記の通り訂正する。

【記】

【国際特許分類第 6 版】

【誤】

G09B 19/05

G06F 9/45

【正】

G05B 19/05

G06F 9/45